

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-162013

(43)Date of publication of application : 19.06.1998

(51)Int.Cl.

G06F 17/30

(21)Application number : 08-317918

(71)Applicant : NIPPON TELEGR & TELEPH CORP <NTT>

(22)Date of filing : 28.11.1996

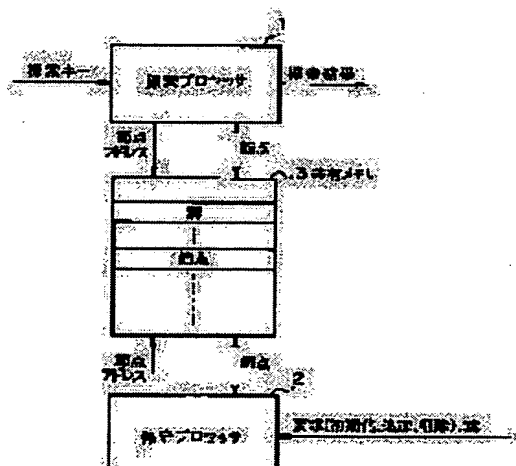
(72)Inventor : TAKAHASHI NAOHISA
MARUYAMA MITSURU
SANEI TAKESHI
OGURA TAKESHI
KAWANO TETSUO
YAGI SATORU

(54) DIGITAL SEARCHING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a digital searching device which executes a searching processing on a digital search tree at a high speed even if the number of headers in the digital search tree is increased, even if the request frequency of the searching processing, an elimination processing and an addition processing increase or even if the requests of the searching processing are continuously outputted.

SOLUTION: In the digital searching device, the digital search tree is kept in a common memory 3 and a maintenance processor 2 executes the initialization processing of the digital search tree, the addition processing of leaves and nodes and the elimination processing of the leaves and the nodes. A searching processor 1 executes the searching processing of the digital search tree. The maintenance processor 2 and the searching processor 1 independently operate and they access to the common memory 3 so as to operate the digital search tree in parallel.



LEGAL STATUS

[Date of request for examination] 21.12.1998

[Date of sending the examiner's decision of rejection] 21.11.2001

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3284064

[Date of registration] 01.03.2002

[Number of appeal against examiner's decision of rejection] 2001-22980

[Date of requesting appeal against examiner's decision of rejection] 21.12.2001

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平10-162013

(43)公開日 平成10年(1998) 6月19日

(51)Int.Cl.⁶

識別記号

F I

G 0 6 F 17/30

G 0 6 F 15/40

3 1 0 B

15/419

3 1 0

審査請求 未請求 請求項の数7 O L (全 15 頁)

(21)出願番号 特願平8-317918

(22)出願日 平成8年(1996)11月28日

(71)出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72)発明者 高橋 直久

東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72)発明者 丸山 充

東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72)発明者 三栄 武

東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(74)代理人 弁理士 若林 忠

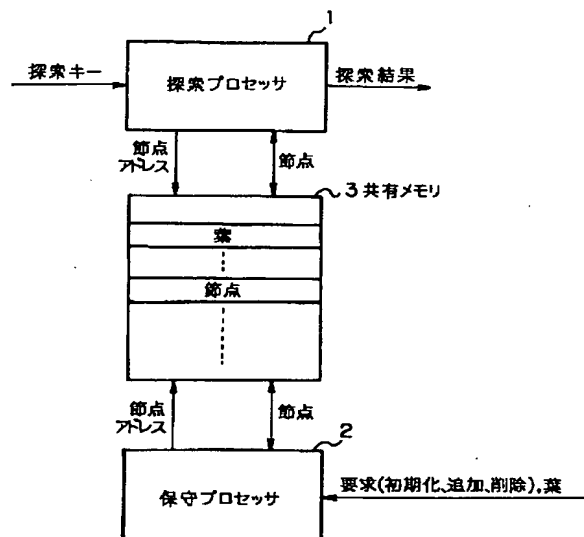
最終頁に続く

(54)【発明の名称】 デジタル探索装置

(57)【要約】

【課題】 デジタル探索木の見出し数の増加、探索処理、削除処理、追加処理の要求頻度の増加に対しても、また、探索処理の要求が連続的に出された場合でも、更に、長い見出しと長い探索キーを扱う場合でも、デジタル探索木に対する探索処理を高速に行なうデジタル探索装置を提供する。

【解決手段】 デジタル探索装置においては、デジタル探索木を共有メモリ3に保持し、デジタル探索木の初期化処理、葉と節点の追加処理、葉と節点の削除処理を保守プロセッサ2が行ない、デジタル探索木の探索処理を探索プロセッサ1が行なう。保守プロセッサ2と探索プロセッサ1は、それぞれ独立に動作し、共有メモリ3をアクセスしてデジタル探索木を並列に操作する。



1

【特許請求の範囲】

【請求項1】 探索したいデータの見出しである探索キーを候補データの見出しと直接比較するのではなく、探索キーを構成している桁を手がかりにして目的データの格納場所に近付いていく、デジタル探索あるいは桁探索と呼ばれるデータ探索装置であって、

デジタル探索木の初期化、前記デジタル探索木の節点と葉の追加と削除、デジタル探索木の再構成とを行なう保守プロセッサと、前記デジタル探索木を探索する探索プロセッサと、前記デジタル探索木を保持する共有メモリとを有し、前記保守プロセッサと前記探索プロセッサがそれぞれ独立に動作し、前記共有メモリをアクセスしてデジタル探索木を並列に操作するデジタル探索装置。

【請求項2】 前記共有メモリが、前記デジタル探索木の節点データとして、検査ビット位置と、探索キーの検査ビット位置のビットの値が0のときに分岐すべき節点アドレスおよび1のときに分岐すべき節点アドレスを保持し、前記探索プロセッサが、前記共有メモリから読み出した節点データに従って、探索キーの前記検査ビット位置のビットの値を調べ、その結果により、次に前記共有メモリから読み出す節点データのアドレスを決定する手段を有し、前記保守プロセッサが、葉の追加の際に、節点データとして前記共有メモリが保持すべき内容が上記の通りになるように、節点データの変更、あるいは、新しい節点を追加して前記デジタル探索木を更新する手段を有する請求項1記載のデジタル探索装置。

【請求項3】 前記共有メモリが、前記デジタル探索木の節点データとして予め定められた数の複数の検査ビット位置と、それらの位置に対する探索キーのビットの値の組合せごとに定まる一連の分岐先節点アドレスを保持し、前記探索プロセッサが、前記共有メモリから読み出した節点データに従って、探索キーの前記複数の検査ビット位置のビットの値を調べ、それらの結果により、次に前記共有メモリから読み出すべき節点データのアドレスを決定する手段を有し、前記保守プロセッサが、葉の追加の際に、検査ビット位置の数が予め定められた値以下になるように、当該節点の節点データの変更、あるいは、新しい節点を追加して前記デジタル探索木を更新する手段を有する請求項1記載のデジタル探索装置。

【請求項4】 前記共有メモリが、前記デジタル探索木の節点データおよび葉のデータとして、探索キーと見出しの比較の際に無視してよいビットを指定するマスクデータをさらに保持し、前記探索プロセッサが、節点データとして前記マスクデータを保持する節点を読み出す際に、その節点データをスタックに積み込む手段と、前記探索木を辿って到達した葉の見出しが探索キーと異なる場合に、前記スタックをポップアップして節点データを取り出す手段と、取り出した節点データのマスクデータを用いて探索キーの一部を無視しながら当該節点からデジタル探索を再度やり直す手段とをさらに有する請求項

2

2または3記載のデジタル探索装置。

【請求項5】 前記共有メモリが前記検査ビット位置(CP)の代わりにCPの移送幅(CP-SKIP)を保持し、前記探索プロセッサが、前記検査ビット位置を示す検査ビット位置レジスタを備えて、前記共有メモリから読み出して得られるCP-SKIPを前記検査ビット位置レジスタの値に足し込むことにより検査ビット位置を決定する手段を有し、前記保守プロセッサが、葉の追加の際に、前記デジタル探索木の既存節点のCP-SKIPの値と予め定められたCP-SKIPの最大値とから、CP-SKIPの値が該最大値以下になるように、ひとつあるいは複数の節点を追加して前記デジタル探索木を更新する手段を有する請求項2から4のいずれか1項に記載のデジタル探索装置。

【請求項6】 前記共有メモリが各節点の節点データとして、分岐先節点である子節点の節点データアドレスと、子節点での分岐先決定のための条件である分岐条件の対からなる枝データを分岐先の数だけ保持し、前記探索プロセッサが前記共有メモリから枝データを読み出す手段と、読み出した枝データの分岐条件と探索キーとから子節点での分岐先節点である孫節点を決定し、子節点の節点データにおける当該孫節点に対する枝データの格納アドレスを計算して、前記共有メモリから次に読み出すべき枝データのアドレスとして設定する手段を有する請求項1から5のいずれか1項に記載のデジタル探索装置。

【請求項7】 前記共有メモリが前記デジタル探索木の葉に対してオン、オフ2値の状態表示データ、および、葉の行列であるLRU行列を表す順序関係を保持し、前記保守プロセッサが前記デジタル探索木に葉を追加する際に、当該葉を前記LRU行列の最後に加えて、その葉の状態表示データをオフにする手段と、一定時間ごと、あるいは指定された契機に、前記LRU行列中の葉の状態表示データを調べ、オンであれば該葉が前記LRU行列の後ろにくるように前記LRU行列の順序を変更する手段と、指定された契機に前記LRU行列の先頭の葉を前記デジタル探索木から削除する手段を有し、前記探索プロセッサが探索キーに対応する葉を探索したときに、その葉の状態表示データをオンにする手段を有する請求項1から6のいずれか1項に記載のデジタル探索装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、探索したいデータの見出し(探索キーという)を候補データの見出しと直接比較するのではなく、探索キーを構成している桁を手がかりにして目的データの格納場所に近付いていく、デジタル探索あるいは桁探索と呼ばれるデジタル探索装置に関する。

【0002】

【従来の技術】 一般に、デジタル探索装置においては、

大別して、

- (1) デジタル探索木の初期化处理
- (2) デジタル探索木の探索処理
- (3) デジタル探索木の葉と節の削除処理
- (4) デジタル探索木への葉と節の追加処理

が行なわれる。

【0003】従来のデジタル探索装置における上記

(1)～(4)の処理について、代表的なデジタル探索木であるパトリシア木を例に用いて説明する。ここでは、デジタル探索木で保持対象となる見出しとして、A 10
～Z、nullの27種類を考える。それぞれの符号の2進数表示を00001～11010、00000とする。符号の2進数表示において、1番左のビットをビット0と呼び、以下左から順に、ビット1、2、3、4と呼ぶ。図8は、null、A、C、E、G、H、I、N、P、R、Sの11個の見出しが存在するパトリシア木である。図で、大きな○は節点、小さな○は葉、節点間、あるいは節点と葉を結ぶ線は枝を表す。節点または葉の上に記した数値は節点または葉を識別するための番号を表し、節点の中に書かれた数値は、その節点で調べ 20
るべき探索キーの桁（検査ビット位置という）を表す。葉の下に記した記号は、その葉の見出しを表す。節点0はデジタル探索木の根といい、根を指している矩形をheadという。パトリシア木では、探索キーが与えられると、探索木の根から順に枝に沿って節点を辿り、探索キーに対して検査ビット位置のビットが0の時は左側の節点に下がり、1の時には右側の節点に下がる。この操作を繰り返すと、その探索キーに等しい見出しの葉が探索木に存在する場合には、その葉に到達できる。たとえば、Hを探索するには、Hの符号01000を探索キー 30
として用いて、木の根の節点から順に、節点が指示する検査ビット位置の探索キーのビットの値を調べて、節点を辿っていけばよい。この場合、最初の節点で探索キーのビット0を調べると0であるので、左下の節点に行く。次の節点でビット1を調べると1なので右下の節点に行く。このように、節点0、1、4、9の順に辿って、それぞれ探索キーのビット0、1、2、4を調べていくと、見出しHの葉に至ることが分かる。

【0004】従来のデジタル探索装置において、初期化处理(1)は以下に行なわれる。

【0005】(a) デジタル探索木のheadを作成する。

【0006】(b) デジタル探索木が空であることをheadに表示する。

【0007】また、従来のデジタル探索装置において、探索処理(2)は以下に行なわれる。

【0008】図8のデジタル探索木で、探索キーがI（符号01001）の葉を探索する場合、次のようになる。

【0009】(c) 探索キーのビット0、1、2がそれ 50

ぞれ0、1、0なので、節点0、1、4の順に辿り、節点9に到達する。節点9で探索キーのビット4を調べると1なので、葉18に到達する。

【0010】(d) 葉18の見出しはIで、探索キーと等しいので、求めるべき葉に到達したことが判明する。このとき、到達した葉（この場合は葉18）の見出しが探索キーと等しくない場合には、探索したい葉がこのデジタル探索木に存在しないことが判明する。

【0011】また、従来のデジタル探索装置において、削除処理(3)は以下に行なわれる。

【0012】図8のデジタル探索木で、見出しI（符号01001）の葉を削除する場合、次のようになる。

【0013】(e) Iの符号を探索キーとして上記(c)と同様に図8の探索木を辿ると、葉18に到達する。

【0014】(f) 葉18の見出しはIで、探索キーと等しいので、削除すべき葉に到達したことが判明する。このとき、到達した葉（この場合は葉18）の見出しが探索キーと等しくない場合には、削除したい葉がこのデジタル探索木に存在しないことが判明する。

【0015】(g) 葉18および葉18に接続された枝を削除する。これにより、節点9の枝が1つになるので、節点9およびその左下の枝を削除して、代わりに葉17（見出しH）を節点4の右下の枝につなげる。

【0016】また、従来のデジタル探索装置において、追加処理(4)は以下に行なわれる。

【0017】図8のデジタル探索木に、見出しK（符号01011）の葉を追加する場合、次のようになる。

【0018】(h) Kの符号を探索キーとして上記(c)と同様に図8の探索木を辿ると、葉18に到達する。

【0019】(i) 葉18の見出しはI（01001）なので、KとIの符号をビット0から順に比べる。この結果、ビット3で、両符号のビットがはじめて異なる値になることが判明する。

【0020】(j) 検査ビット位置が3である節点23を新たに作成し、節点4と節点9の間に挿入する。

【0021】(k) 節点23の右下側の枝に新たに葉23を作成し、その見出しをKにする。

【0022】

【発明が解決しようとする課題】上記従来技術においては、デジタル探索木に対する初期化处理、探索処理、削除処理、追加処理を単一のプロセッサで行なっていたため、下記のような問題があった。

【0023】(1) 探索処理、削除処理、追加処理の要求が連続して多数来たとき、削除処理、追加処理にCPU時間をとられ、探索処理が遅延することがある。

【0024】(2) 探索処理、削除処理、追加処理の要求が同時に来ることがあるような応用において、要求を逐次的に処理しなければならないので、探索処理が後回

5

しにされて遅延することがある。

【0025】(3) 探索処理の高速化のために、プロセッサを処理に合わせて専用化したい場合でも、削除処理、追加処理が複雑であるため、プロセッサの構造が複雑になり実現困難である。

【0026】(4) 探索処理では、デジタル探索木において、1つの節点で探索キーのビットを1つ調べて、節点を順に辿らなければならない。このため、見出し数が多いデジタル探索木の場合、あるいは、見出しの長さが長い場合に、辿るべき節点の数が多いので、探索処理時間10
時間が長くなることがある。

【0027】(5) デジタル探索木の節点のデータとして検査ビット位置を用いているので、見出しが長い場合に検査ビット位置のデータも長くなるので、探索処理で1つの節点を辿るためにメモリから読み込まなければならないデータ量が多くなり、探索処理が遅くなる。

【0028】(6) 探索処理では、検査ビット位置を読み出して、探索キーを調べた後で、始めて次節点アドレスの格納場所を特定することができる。このため、節点を1つ辿る際に、検査ビット位置の読み出しと、次節点20
アドレスの読み出しの2回のメモリアクセスを逐次的に行なわなければならないので、処理時間が長くなる。検査ビット位置を読み出す時点では、次節点アドレスの候補は複数存在するので、1度のメモリアクセスで済ますためには、それら複数の次節点アドレスを同時に読み出さなければならない。見出し数が多い場合には、次節点アドレスも長くなるので、このことは、プロセッサとメモリ間のデータ転送用のバス幅を実現困難なほど広くしなければならないことを意味する。

【0029】また、デジタル探索木の各節点ごと、あるいはいくつかまとめた節点ごとに、探索処理、削除処理、追加処理を行なうプロセッサを割り付けて、複数のプロセッサでこれらの処理を行なう提案例もあるが、これにも下記のような問題があった。

【0030】(7) デジタル探索木の構成に応じて、プロセッサの数などのシステムの構成を変えなければならない。特に、見出し数の多いデジタル探索木の場合には、システムの規模が拡大し、かつ複雑になってしまう。

【0031】(8) 節点を移動するごとにプロセッサ間40
通信が発生するので、通信のオーバーヘッドが大となる。

【0032】また、上記いずれの提案例、実現例においても、以下のような問題がある。

【0033】(9) デジタル探索木に探索キーと一致する見出しを持つ葉がない場合に、必ず探索失敗（該当する見出しなし）となってしまう、特定の一部のビットだけが一致するような見出しを探索結果として求められるようにできない。

【0034】本発明の目的は、デジタル探索木の見出し数の増加、探索処理、削除処理、追加処理の要求頻度の50

6

増加に対しても、また、探索処理の要求が連続的に出された場合でも、更に、長い見出しと長い探索キーを扱う場合でも、デジタル探索木に対する探索処理を高速に行なうデジタル探索装置を提供することである。

【0035】

【課題を解決するための手段】本発明の上記目的は、デジタル探索木の初期化、デジタル探索木の節点と葉の追加と削除、デジタル探索木の再構成を行なう保守プロセッサと、デジタル探索木を探索する探索プロセッサと、デジタル探索木を保持する共有メモリとを有し、保守プロセッサと探索プロセッサがそれぞれ独立に動作し、共有メモリをアクセスしてデジタル探索木を並列に操作するデジタル探索装置により達成される。本発明のデジタル探索装置においては、デジタル探索木を共有メモリに保持し、デジタル探索木の初期化処理、葉と節点の追加処理、葉と節点の削除処理を保守プロセッサが行ない、デジタル探索木の探索処理を探索プロセッサが行なう。保守プロセッサと探索プロセッサは、それぞれ独立に動作し、共有メモリをアクセスしてデジタル探索木を並列に操作する。

【0036】したがって、下記の問題が解消される。

【0037】(1) 探索処理、削除処理、追加処理の要求が連続して多数来たとき、削除処理、追加処理にCPU時間をとられ、探索処理が遅延することがある。

【0038】(2) 探索処理、削除処理、追加処理の要求が同時に来ることがあるような応用において、要求を逐次的に処理しなければならないので、探索処理が後回しにされて遅延することがある。

【0039】(3) 探索処理の高速化のために、プロセッサを処理に合わせて専用化したい場合でも、削除処理、追加処理が複雑であるため、プロセッサの構造が複雑になり実現困難である。

【0040】請求項2の実施態様によれば、共有メモリが、デジタル探索木の節点データとして、検査ビット位置と、探索キーの検査ビット位置のビットの値が0のときに分岐すべき節点アドレスおよび1のときに分岐すべき節点アドレスを保持し、探索プロセッサが、共有メモリから読み出した節点データに従って、探索キーの検査ビット位置のビットの値を調べ、その結果により、次に共有メモリから読み出す節点データのアドレスを決定する手段を有し、保守プロセッサが、葉の追加の際に、節点データとして共有メモリが保持すべき内容が上記の通りになるように、節点データの変更、あるいは、新しい節点を追加してデジタル探索木を更新する手段を有する。

【0041】請求項3の実施態様によれば、共有メモリが、デジタル探索木の節点データとして、予め定められた数の複数の検査ビット位置と、それらの位置に対する探索キーのビットの値の組合せごとに定まる一連の分岐先節点アドレスを保持し、前記探索プロセッサが、前記

共有メモリから読み出した節点データに従って、探索キーの前記複数の検査ビット位置のビットの値を調べ、それらの結果により、次に前記共有メモリから読み出すべき節点データのアドレスを決定する手段を有し、前記保守プロセッサが、葉の追加の際に、検査ビット位置の数が予め定められた値以下になるように、当該節点の節点データの変更、あるいは、新しい節点を追加してデジタル探索木を更新する手段を有する。

【0042】請求項3においては、探索プロセッサで、探索キーに対して1つの節点で複数のビットを調べるこ

【0043】探索処理では、デジタル探索木において、1つの節点で探索キーのビットを1つ調べて、節点を順に辿らなければならないため、見出し数が多いデジタル探索木の場合、あるいは、見出しの長さが長い場合に、辿るべき節点の数が多いので、探索処理時間が長くなることがある。

【0044】請求項4の実施態様によれば、共有メモリが、デジタル探索木の節点データおよび葉のデータとして、探索キーと見出しの比較の際に無視してよいビットを指定するマスクデータをさらに保持し、探索プロセッサが、節点データとしてマスクデータを保持する節点を読み出す際に、その節点データをスタックに積み込む手段と、探索木を辿って到達した葉の見出しが探索キーと異なる場合に、前記スタックをポップアップして節点データを取り出す手段と、取り出した節点データのマスクデータを用いて探索キーの一部を無視しながら当該節点からデジタル探索を再度やり直す手段とをさらに有する。

【0045】請求項4においては、節点データと葉のデータとしてマスクデータを用いて探索キーの一部を無視できるようにしており、探索プロセッサがスタックを用いてマスクデータを保持している節点データへの到着履歴を管理して、探索キーと異なる葉に到達した場合にバックトラックできるようにしているので以下の問題が解消する。

【0046】デジタル探索木に探索キーと一致する見出しを持つ葉がない場合に、必ず探索失敗（該当する見出しなし）となってしまう、特定の一部のビットだけが一致するような見出しを探索結果として求められるようにできない。

【0047】請求項5の実施態様によれば、共有メモリが前記検査ビット位置（CP）の代わりにCPの移動幅（CP-SKIP）を保持し、前記探索プロセッサが、検査ビット位置を示す検査ビット位置レジスタを備えて、前記共有メモリから読み出して得られるCP-SKIPを検査ビット位置レジスタの値に足し込むことにより検査ビット位置を決定する手段を有し、前記保守プロセッサが、葉の追加の際に、デジタル探索木の既存節点

のCP-SKIPの値、および、予め定められたCP-SKIPの最大値とから、CP-SKIPの値が該最大値以下になるように、ひとつあるいは複数の節点を追加してデジタル探索木を更新する手段を有する。

【0048】請求項5においては、節点のデータとして、検査ビット位置の代わりに見出しの長さに依存しない短い長さに抑えることが可能な検査ビット移動幅を用いているので、以下の問題が解消できる。

【0049】デジタル探索木の節点のデータとして検査ビット位置を用いているので、見出しが長い場合に検査ビット位置のデータも長くなるので、探索処理で1つの節点を辿るためにメモリから読み込まなければならないデータ量が多くなり、探索処理が遅くなる。

【0050】請求項6の実施態様によれば、共有メモリが各節点の節点データとして、分岐先節点である子節点の節点データアドレスと、子節点での分岐先決定のための条件である分岐条件の対からなる枝データを分岐先の数だけ保持し、探索プロセッサが共有メモリから枝データを読み出す手段と、読み出した枝データの分岐条件と探索キーとから子節点での分岐先節点である孫節点を決

【0051】請求項6においては、探索プロセッサが節点を1つ辿るためにメモリから読み込まなければならないデータは1つの節点アドレスと分岐条件だけであり、さらに、これらは同時に読み込むことができるので、次の問題を解消できる。

【0052】（1）探索処理では、検査ビット位置を読み出して、探索キーを調べた後で、始めて次節点アドレスの格納場所を特定することができる。このため、節点を1つ辿る際に、検査ビット位置の読み出しと、次節点アドレスの読み出しの2回のメモリアクセスを逐次的に行なわなければならないので、処理時間が長くなる。検査ビット位置を読み出す時点では、次節点アドレスの候補は複数存在するので、1度のメモリアクセスで済ますためには、それら複数の次節点アドレスを同時に読み出さなければならない。見出し数が多い場合には、次節点アドレスも長くなるので、このことは、プロセッサとメモリの間のデータ転送用のバス幅を実現困難なほど広くしなければならないことを意味する。

【0053】さらに、探索プロセッサと、保守プロセッサと、共有メモリで装置を構成し、各プロセッサは、それぞれ独立にデジタル探索木を辿ればよいので、以下の問題は発生しない。

【0054】（2）デジタル探索木の構成に応じて、プロセッサの数などのシステムの構成を変えなければならない。特に、見出し数の多いデジタル探索木の場合には、システムの規模が拡大し、かつ複雑になってしま

う。

【0055】(3) 節点を移動するごとにプロセッサ間通信が発生するので、通信のオーバーヘッドが大となる。

【0056】請求項7の実施態様によれば、共有メモリがデジタル探索木の葉に対してオン、オフ2値の状態表示データ、および、葉の行列であるLRU行列を表す順序関係を保持し、前記保守プロセッサがデジタル探索木に葉を追加する際に、当該葉をLRU行列の最後に加えて、その葉の状態表示データをオフにする手段と、一定時間ごと、あるいは指定された契機に、LRU行列中の葉の状態表示データを調べ、オンであれば該葉数がLRU行列の後にくるようにLRU行列の順序を変更する手段と、指定された契機にLRU行列の先頭の葉をデジタル探索木から削除する手段を有し、前記探索プロセッサが探索キーに対応する葉を探索したときに、その葉の状態表示データをオンにする手段を有する。

【0057】このように、LRU行列で葉の生成順序を保持し、オン/オフの状態表示データで探索プロセッサでの探索状況を保持することにより、作られた順が古く、かつ最近探索されていない葉から順番に削除対象として選ばれる。また、作られた順が古くても、ある一定時間前からの間に一度でも探索された葉は状態がオンになるので、他に状態がオフの葉がある限り削除対象の候補から除外される、あるいは、候補になりにくくなる。

【0058】

【発明の実施の形態】次に、本発明の実施形態について図面を参照して説明する。

【0059】図1は本発明の一実施形態のデジタル探索装置の構成図である。

【0060】本実施形態のデジタル探索装置は、デジタル探索木の初期化、デジタル探索木の節点と葉の追加と削除、デジタル探索木の再構成とを行なう保守プロセッサ2と、デジタル探索木を探索する探索プロセッサ1と、デジタル探索木を保持する共有メモリ3とから構成され、保守プロセッサ2と探索プロセッサ1がそれぞれ独立に動作し、共有メモリ3をアクセスしてデジタル探索木を並列に操作する。デジタル探索木の初期化処理、追加処理、削除処理の要求は保守プロセッサ2に送られる。また、探索処理の要求は、探索プロセッサ1に送られる。追加処理の要求では、追加すべき葉の見出しと、その見出しに結びつけられたデータ(見出し対応データと呼ぶ)の格納アドレスが与えられる。削除処理の要求では、削除すべき葉の見出しが与えられる。探索処理では、探索すべき葉の見出しが与えられる。

【0061】請求項2に対応する第1の実施形態では、共有メモリ3に、デジタル探索木の節点データとして、図8に例示したと同様に、検査ビット位置と、探索キーの検査ビット位置のビットの値が0のときに分岐すべき

$$b0 = s1 \quad \dots \quad (1)$$

$$b1 = (s1 \cdot s2) \vee (\text{not}(s1) \cdot s2) \dots \quad (2)$$

節点アドレスおよび1のときに分岐すべき節点アドレスを保持し、探索プロセッサ1は、共有メモリ3から読み出した節点データに従って、探索キーの検査ビット位置のビットの値を調べ、その結果により、次に共有メモリ3から読み出す節点データのアドレスを決定する手段

(不図示)を有し、保守プロセッサ2は、葉の追加の際に、節点データとして共有メモリ3が保持すべき内容が上記の通りになるように、節点データの変更、あるいは、新しい節点を追加してデジタル探索木を更新する手段(不図示)を有する。

【0062】請求項3に対応する第2の実施形態では、共有メモリ3が、デジタル探索木の節点データとして、複数の検査ビット位置と、それらの位置に対する探索キーのビットの値の組合せごとに定まる一連の分岐先節点アドレスを保持し、探索プロセッサ1が、共有メモリ3から読み出した節点データに従って、探索キーの複数の検査ビット位置のビットの値を調べ、それらの結果により、次に共有メモリ3から読み出すべき節点データのアドレスを決定する手段(不図示)を有し、保守プロセッサ2が、葉の追加の際に、検査ビット位置の数が予め定められた値以下になるように、当該節点の節点データの変更、あるいは、新しい節点を追加してデジタル探索木を更新する手段(不図示)を有する。

【0063】デジタル探索木において、根の節点は第0段にあるといい、根から出た枝に接続された節点は第1段にあるといい、同様に、第i段の節点から出た枝に接続された節点は第(i+1)段にあるというように、節点に対して、根から辿るときに通る枝の数で表される段に、その節点はあるという。図2は、図8において、連続した2つの段にある節点を、4本の枝を持つ1つの節点にまとめることによりできるデジタル探索木である。この図の記法は図8と同様である。ただし、節点の中に記した3つの数値の並びは、3つの検査ビット位置を表す。ここで、節点1、2、3、5のように、3つの数値の並びのなかに同じ数値がある場合には、その節点での検査ビット位置の数が2以下であることを表す。

【0064】また、各節点は4本の枝を持ち、左から順に、第0、第1、第2、第3の枝と呼ぶ。

【0065】探索プロセッサ1と保守プロセッサ2は、探索処理、追加処理、削除処理において、これらの検査ビットと4本の枝を使って、以下のように節点を辿ると、図8のデジタル探索木を、第1の実施形態の処理で辿った場合と同じ見出しを持つ葉に到達できる。今、検査ビット位置がCP1、CP2、CP3とする。このとき、探索キーのビットCP1、CP2、CP3がそれぞれs1、s2、s3である場合には、以下の式で計算されるbの値を用いて、第bの枝を辿る。

【0066】

11

 $b = b1 \& b0$

ここで、「・」、「v」は、それぞれ2つのビットの論理積、論理和を、「not (s1)」はs1の否定を表し、「&」はビットの接続を表し、 $b = b1 \& b0$ の式は、下位ビットb0、上位ビットb1の2ビットでbが構成されていることを表す。

【0067】探索プロセッサ1において、探索処理は以下のように行なわれる。

【0068】図2のデジタル探索木で、探索キーがI (符号01001) の葉を探索する場合、次のようにな 10
る。

【0069】(3c) 探索キーのビット0、1、2がそれぞれ0、1、0なので、節点0で第1の枝が選ばれて、節点2に到達する。節点2で探索キーのビット2、4を調べると0、1なので、第1の枝が選ばれて、葉10に到達する。

【0070】(3d) 葉10の見出しはIで、探索キーと等しいので、求めるべき葉に到達したことが判明する。このとき、到達した葉(この場合は葉10)の見出しが探索キーと等しくない場合には、探索したい葉がこ 20
のデジタル探索木に存在しないことが判明する。

【0071】また、保守プロセッサ2において、削除処理は以下のように行なわれる。

【0072】図2のデジタル探索木で、見出しI (符号01001) の葉を削除する場合、次のようになる。

【0073】(3e) Iの符号を探索キーとして上記(3c)と同様に図2の探索木を辿ると、葉10に到達する。

【0074】(3f) 葉10の見出しはIで、探索キーと等しいので、削除すべき葉に到達したことが判明す 30
る。このとき、到達した葉(この場合は葉10)の見出しが探索キーと等しくない場合には、削除したい葉がこのデジタル探索木に存在しないことが判明する。

【0075】(3g) 葉10および葉10に接続された枝を削除する。さらに、節点2の検査ビット位置の並びを、2、2、2に変更する。

【0076】また、保守プロセッサ2において、追加処理は以下のように行なわれる。

【0077】図2のデジタル探索木に、見出しk (符号01011) の葉を追加する場合、次のようになる。 40

【0078】(3h) kの符号を探索キーとして上記(3c)と同様に図2の探索木を辿ると、葉10に到達する。

【0079】(3i) 葉10の見出しはI (01001) なので、KとIの符号をビット0から順に比べる。この結果、ビット3で、両符号のビットがはじめて異なる値になることが判明する。

【0080】(3j) 節点2の検査ビット位置の並びが2、4、2であるので、まず、これを2、3、2に変更し、この第1の枝に見出しKを持つ葉を接続する。 50

12

... (3)

【0081】(3k) 検査ビット位置の並びが4、4、4である節点を作成し、その第0の枝に見出しHの葉を、第3の枝に見出しIの枝を接続し、この節点を節点2の第0の枝に接続する。

【0082】上記と同様にして、図8において連続した3つの段にある節点を、8本の枝を持つ1つの節点にまとめることによりできるデジタル探索木に対しても処理手順を定めることができる。さらに、図8において、まとめる段の数を増やしていても同様である。

【0083】請求項4に対応する第3の実施形態では、共有メモリ3が、デジタル探索木の節点データおよび葉のデータとして、探索キーと見出しの比較の際に無視してよいビットを指定するマスクデータをさらに保持し、探索プロセッサ1が、節点データとしてマスクデータを保持する節点を読み出す際に、その節点データをスタックに積み込む手段(不図示)と、探索木を辿って到達した葉の見出しが探索キーと異なる場合に、前記スタックをポップアップして節点データを取り出す手段(不図示)と、取り出した節点データのマスクデータを用いて探索キーの一部を無視しながら当該節点からデジタル探索を再度やり直す手段(不図示)とをさらに有する。

【0084】図3は、図8のデジタル探索木の各節点と葉において、上記マスクデータ(以下単にmaskという)を保持したデジタル探索木の例である。デジタル探索装置においては、探索処理は以下のように行なわれる。図3のデジタル探索木で、探索キーが符号10111の場合には、次のように探索する。

【0085】(4c) headから節点0に到達する。節点0はmaskがあるので節点0の節点データをスタックに積む。次に、探索キーのビット0が1なので、節点0から2に行く。節点2はマスクデータを保持しているので節点2の節点データをスタックに積む。次に、探索キーのビット2が1なので、節点2から葉6に到達する。

【0086】(4d-1) 葉6の見出しはT (10100) で、探索キーと等しくないことが判明する。スタックの一番上にある節点2のデータをポップアップする。mask=11000なので、探索キー(10111)を10000に変更する。すなわち、探索キーとmaskの各ビットごとに論理積をとってできるビットパターンを新たな探索キーにする。

【0087】(4d-2) 新しい探索キーのビット2が0なので、節点5に行き、更にビット3も0なので、葉11に到達する。葉11はmask=11000を保持しているので、葉11の見出し(10000)と該maskのビットごとの論理積をとってできるビットパターン10000を探索キーと比較する。

【0088】(4d-3) 両者が一致するので、葉11を探索結果と判定する。

【0089】また、図3のデジタル探索木で、探索キーが符号11101の場合には、次のように探索する。

【0090】(5c)(4c)と同じ。

【0091】(5d-1)(4d-1)と同じ。

【0092】(5d-2)(4d-2)と同じ。

【0093】(5d-3)両者が一致しないので、スタックの一番上にある節点0のデータをポップアップする。mask=00000なので、探索キー(10111)を00000に変更する。

【0094】(5d-4)新しい探索キーのビット0、1、2、3が0なので、節点0、1、3、7の順に辿り節点13に到達する。節点13で探索キーのビットを調べると0なので葉21に到達する。葉21はmask=00000を保持しているため、葉21の見出し(00000)と該maskとビットごとの論理積をとってできるビットパターン00000を探索キーと比較する。

【0095】(5d-5)両者が一致するので、葉21を探索結果と判定する。

【0096】請求項5に対応する第4の実施形態では、共有メモリ3が前記検査ビット位置(CP)の代わりにCPの移動幅(CP-SKIP)を保持し、探索プロセッサ1が、検査ビット位置を示す検査ビット位置レジスタ(不図示)を備えて、共有メモリ3から読み出して得られるCP-SKIPを検査ビット位置レジスタの値に足し込むことにより検査ビット位置を決定する手段(不図示)を有し、保守プロセッサ2が、葉の追加の際に、デジタル探索木の既存節点のCP-SKIPの値、および、予め定められたCP-SKIPの最大値(Mとする)とから、CP-SKIPの値がM以下になるように、ひとつあるいは複数の節点を追加してデジタル探索木を更新する手段(不図示)を有する。

【0097】図4は、図8のデジタル探索木の各節点において、上記に従って、検査ビット位置の代わりに、検査ビット位置の移動幅を持たせてできるデジタル探索木である。

【0098】探索プロセッサ1は、検査ビット位置を示す検査ビット位置レジスタを備えて、共有メモリ3から読み出して得られるCP-SKIPを検査ビット位置レジスタに足し込むことにより検査ビット位置を決定する。保守プロセッサ2は、節点の追加の際に、その節点での検査ビット位置CPから、探索木の根から当該節点に至る経路上の節点のCP-SKIPの値を減じていくことにより、当該節点のCP-SKIPの値を計算する(計算結果をSとする)。Sと予め定められたCP-SKIPの最大値Mと比べて、CP-SKIPの値がM以下になるように、Sの値に応じてひとつあるいは複数の節点を作り出して追加する。たとえば、Sが20、Mが8のとき、CP-SKIPの値が、8、8、4の3つの節点a、b、cを作り、aからb、bからcへ枝を張って、これら3つの節点をデジタル探索木に挿入する。

【0099】次に、探索キーがI(符号01001)の葉を探索する例を用いて、図4のデジタル探索木での探索処理を説明する。まず、検査ビット位置レジスタ(CPR)を0に初期化する。次に、節点0で、CP-SKIPの値が0なので、CPRにCP-SKIPを加えてもCPR=0のままである。このため、探索キーのビット0を調べる。このビットの値が0であるので、次に、左側の節点1に行く。節点1のCP-SKIPの値1をCPRに加えるとCPRは1になるので、探索キーのビット1を調べる。このビットの値が1なので、右側節点4に行く。節点4では、CP-SKIPの値1をCPRに加えて、CPRを2にして、探索キーのビット2を調べる。次に、探索キーのビット2が0なので、節点9に行き、CPRにCP-SKIPの値2を加えて4にして、探索キーのビット4を調べる。このビットの値が1なので、葉18に到達する。ここで、葉18の見出し(I)が探索キーと等しいので、葉18が求める葉であることが分かる。

【0100】請求項6に対応する第5の実施形態では、共有メモリ3が各節点の節点データとして、分岐先節点である子節点の節点データアドレスと、子節点での分岐先決定のための条件である分岐条件の対からなる枝データを分岐先の数だけ保持し、探索プロセッサ1が共有メモリ3から枝データを読み出す手段(不図示)と、読み出した枝データの分岐条件と探索キーとから子節点での分岐先節点である孫節点を決定し、子節点の節点データにおける当該孫節点に対する枝データの格納アドレスを計算して、共有メモリ3から次に読み出すべき枝データのアドレスとして設定する手段(不図示)を有する。

【0101】共有メモリ3は、図6に例示したデジタル探索木を保持する。通常の木構造のデータを保持する場合と同様に、図における枝は、枝の下にある節点または葉のデータを保持しているメモリのアドレスとして表される。このとき、節点として、0~3の4つの枝に対する枝データを、順に格納した4語のデータを保持する。headを保持するメモリアドレスだけが固定(たとえばアドレス0)であり、節点は、上記4語が連続していれば、メモリの空いている任意の場所に保持してよい。また、葉として、見出しと、見出し対応データの格納アドレスの対からなる葉データを保持する。ひとつの葉データはメモリに連続した領域に保持する。葉データの見出しの長さは任意でも実現できる。

【0102】図5は図1のデジタル探索木に対して、上記のように節点データとheadの枝データを保持してできるデジタル探索木である。また、図6は、図2のデジタル探索木に対して、上記のように、節点データとheadの枝データを保持してできるデジタル探索木である。これらの図で、[a, b, c]は枝データを表し、[-]は子節点での分岐条件がない、すなわち、分岐先節点が葉であることを表す。また、単に-と記している

のは、枝データが空、すなわち、枝の先に節点または葉が接続されていないことを表す。

【0103】探索プロセッサ1は、共有メモリ3から枝データを読み出し、読み出した枝データの分岐条件と探索キーとから子節点での分岐先節点(孫節点)を決定し、子節点の節点データにおける当該孫節点に対する枝データの格納アドレスを計算して、共有メモリ3から次に読み出すべき枝データのアドレスとして設定する。

【0104】図6のデジタル探索木で、探索キーがI(符号01001)の葉を探索する場合、次のようになる。

【0105】(6c) headの枝データの分岐条件[0, 1, 2]を読み出す。探索キーのビット0, 1, 2がそれぞれ0, 1, 0なので、節点0の第1の枝データを選んで、読み出す。この枝データは分岐条件が[2, 4, 2]、子節点の節点データアドレスが節点2のアドレスを保持している。このため、探索キーのビット2, 4を調べると0, 1なので、節点2の第1の枝データが選ばれて、分岐条件が[-]なので葉10に到達する。

【0106】上記操作では、節点を1つ辿るために枝データを1つ読み出せばよい。また、枝データにおける、分岐条件と節点アドレスは同時に読み出すことが可能である。すなわち、1回のメモリアクセスで節点を1つ辿ることができる。

【0107】(6d) 葉10の見出しはIで、探索キーと等しいので、求めるべき葉に到達したことが判明する。このとき、到達した葉(この場合は葉10)の見出しが探索キーと等しくない場合には、探索したい葉がこのデジタル探索木に存在しないことが判明する。

【0108】また、保守プロセッサ2において、削除処理は以下に行なわれる。

【0109】図6のデジタル探索木で、見出しI(符号01001)の葉を削除する場合、次のようになる。

【0110】(6e) Iの符号を探索キーとして上記(4c)と同様に図5の探索木を辿ると、葉10に到達する。

【0111】(6f) 葉10の見出しはIで、探索キーと等しいので、削除すべき葉に到達したことが判明する。このとき、到達した葉(この場合は葉10)の見出しが探索キーと等しくない場合には、削除したい葉がこのデジタル探索木に存在しないことが判明する。

【0112】(6g) 葉10および葉10に接続された枝を削除する。さらに、節点2の第1の枝データを-に変更する。

【0113】また、保守プロセッサ2において、追加処理は以下に行なわれる。

【0114】図6のデジタル探索木に、見出しK(符号01011)の葉を追加する場合、次のようになる。

【0115】(6h) Kの符号を探索キーとして上記

(6c)と同様に図6の探索木を辿ると、葉10に到達する。

【0116】(6i) 葉10の見出しはI(01001)なので、KとIの符号をビット0から順に比べる。この結果、ビット3で、両符号のビットがはじめて異なる値になることが判明する。

【0117】(6j) 節点0の第1の枝データの分岐条件[2, 4, 2]であるので、まず、これを[2, 3, 2]に変更する。さらに、節点0の第1の枝に接続されている節点2の第1の枝に見出しKを持つ葉を接続する。

【0118】(6k) 節点2の第0の枝データの分岐条件を[4, 4, 4]に変更する。分岐条件が、[-], -, -, [-]の節点を作成し、その第0の枝に見出しHの葉を、第3の枝に見出しIの葉を接続し、この節点を節点2の第0の枝に接続する。

【0119】以下では簡単のため枝データの長さ(1語)の倍数の固定長に設定し、見出し対応データ格納アドレスは1語に設定した場合について説明する。

【0120】探索プロセッサ1は、探索要求を受け取り、探索キーが与えられると、まず、上記(6c)に例示した手順に従い、葉に到達するまで次の操作を繰り返す。

【0121】・共有メモリ3から枝データを読み出す。

【0122】・枝データから分岐条件と分岐先節点アドレスを取り出す。

【0123】・前記(式1)～(式3)を分岐条件と探索キーに適用して、枝番号を求める。

【0124】・分岐先節点アドレスに枝番号を加算することにより、次に読み出す枝データのアドレスを求める。

【0125】次に、上記(6d)に例示した手順に従い、次の操作を行ない、探索結果を作成する。

【0126】・共有メモリ3から葉データの先頭語を読み出す。

【0127】・先頭語を探索キーの先頭語と比較して、異なっている場合には該当する葉は存在しないと出力して終了する。

【0128】・一致した場合には、共有メモリ3から葉データの第2語を読み出す。

【0129】・読み出した語を探索キーの第2語と比較して、異なっている場合には該当する葉は存在しないと出力して終了する。

【0130】・上記操作を見出しと探索キーのすべての語について行ない、見出しと探索キーが等しいか調べる。

【0131】・見出しと探索キーが等しい場合には、共有メモリ3から次の語を読み出すことにより、見出し対応データ格納アドレスを得て、探索結果として返す。

【0132】保守プロセッサ2は、削除処理の要求を受

け取り、削除すべき葉の見出しを与えられると、前記(6e)～(6g)の手順で、共有メモリ3に保持されているデジタル探索木を変更して、葉データを削除する。

【0133】また、保守プロセッサ2は、追加処理の要求を受け取り、追加すべき葉の見出しと見出し対応データ格納アドレスを与えられると、前記(6h)～(6k)の手順で、共有メモリ3に保持されているデジタル探索木を変更して、葉データを追加する。

【0134】請求項7に対応する第6の実施形態では、共有メモリ3がデジタル探索木の葉に対して、オン、オフ2値の状態表示データ、および、葉の行列であるLRU行列を表す順序関係を保持し、保守プロセッサ2がデジタル探索木に葉を追加する際に、当該葉をLRU行列の最後に加えて、その葉の状態表示データをオフにする手段(不図示)と、一定時間ごと、あるいは指定された契機に、LRU行列中の葉の状態表示データを調べ、その値に従ってLRU行列の順序を変更する手段(不図示)と、指定された契機にLRU行列の先頭の葉をデジタル探索木から削除する手段(不図示)を有し、探索プロセッサ1が探索キーに対応する葉を探索したときに、その葉の状態表示データをオンにする手段(不図示)を有する。

【0135】保守プロセッサ2は、デジタル探索木に葉を追加する際に、当該葉をLRU行列の最後に加えて、その葉の状態表示データをオフにする。さらに、一定時間ごと、あるいは、指定された契機に、LRU行列中の葉の状態表示データを調べ、その値に従ってLRU行列の順序を変更する。また、指定された契機にLRU行列の先頭の葉をデジタル探索木から削除する。探索プロセッサ1は、探索キーに対応する葉を探索したときに、その葉の状態表示データをオンにする。

【0136】次に、図7に示すデジタル探索木構成例を用いて、請求項8に対応する第6の実施形態における保守プロセッサ2の動作を説明する。図で、点線の矢印はLRU行列における順序関係を表すものである。これは、たとえば、矢印の元側で、矢印の先にある葉のアドレスを保持することにより実現できる。また、葉の下に記入したオン、オフは状態表示データの値を示す。

【0137】保守プロセッサ2は、新たに葉を追加しようとしたとき、葉の追加に必要なメモリがない場合には、デジタル探索木に既に存在している葉を削除して必要なメモリを作り出さなければならない。この場合に、LRU行列の先頭から順に葉を辿り、状態がオンならば、LRU行列の最後に移し、オフならば、デジタル探索木から削除する。図7では、LRUfirstから辿り、最初の2つの葉、葉4(T)、葉14(S)は状態がオンなので、葉15の後ろに移し、3番目の葉、葉13(R)は状態がオフなので、削除する。さらに、葉12(P)、葉11(N)、葉7(E)、葉16(A)の

状態がオンなので、これらの葉を順にLRU行列の後ろに移す。このとき、これら後ろに移した葉の状態をオフに変更する。この結果、LRU行列は、葉10(1)を先頭に、9、8、6、15、4、14、12、11、7、16の順になる。

【0138】このように、LRU行列で葉の生成順序を保持し、オン/オフの状態表示データで探索プロセッサ1での探索状況を保持することにより、作られた順が古く、かつ最近探索されていない葉から順番に削除対象として選ばれる。また、作られた順が古くても、ある一定時間前からの間に一度でも探索された葉は状態がオンになるので、他に状態がオフの葉がある限り削除対象の候補から除外される、あるいは、候補になりにくくなる。

【0139】

【発明の効果】以上説明したように、本発明は下記のような効果がある。

(1) 請求項1の発明は、プロセッサをデジタル探索木の探索処理を専門に行なう探索プロセッサとその他の処理を行なう保守プロセッサに分離して、探索処理に対する要求を、その他の処理とは独立に並行して受け取って処理することにより、

①探索処理、削除処理、追加処理の要求が連続して多数来たときに、削除処理、追加処理にCPU時間をとられ、探索処理が遅延することがない。

【0140】②探索処理、削除処理、追加処理の要求が同時に来ることがあるような応用において、探索処理が後回しにされて遅延することがない。

【0141】③探索処理の高速化のために、プロセッサを処理に合わせて専用化したい場合でも、プロセッサの構造が簡単で、実現可能である。

(2) 請求項2の発明は、探索処理を簡単にして、かつ、削除処理、追加処理と独立させているので、高速な探索処理プロセッサを容易に実現できる。このため、特に、削除処理、追加処理に比べて、探索処理の頻度が高い応用に対して、価格性能比の高いデジタル探索装置を作ることができる。

(3) 請求項3の発明は、探索プロセッサで、探索キーに対して1つの節点で複数のビットを調べることが可能になるようにしたため、

①見出し数が多いデジタル探索木の場合、あるいは、見出しの長さが長い場合に、探索処理時間が長くなることがない。

(4) 請求項4の発明は、節点データと葉のデータとしてマスクデータを用いて探索キーの一部を無視できるようにしており、探索プロセッサがスタックを用いてマスクデータを保持している節点データへの到着履歴を管理して、探索キーと異なる葉に到達した場合にバックトラックできるようにしているので、デジタル探索木に探索キーと一致する見出しを持つ葉がない場合に、特定の一部のビットだけ一致するような見出しを探索結果として

求めることができる。

(5) 請求項5の発明は、節点のデータとして、検査ビット位置の代わりに見出しの長さに依存しない短い長さに抑えることが可能な検査ビット移動幅を用いているので、見出しが長い場合にも、探索処理で1つの節点を辿るためにメモリから読み込まなければならないデータ量を少なく抑えることができる。また、プロセッサとメモリの間のデータ転送用のバス幅を小さく抑えることができる。

(6) 請求項6の発明は、探索プロセッサが節点を1つ 10 辿るためにメモリから読み込まなければならないデータは、1個の節点アドレスと分岐条件だけであり、さらに、これらは同時に読み込むことができるので、①見出し数が多い場合にも、プロセッサとメモリの間のデータ転送用のバス幅を実現困難なほど広くする必要がない。

【0142】②デジタル探索木の構成に応じて、プロセッサの数などのシステムの構成を変えなくてもよい。特に、見出し数の多いデジタル探索木の場合にも、システムの規模が拡大し、かつ複雑になることがない。 20

【0143】③節点を移動してもプロセッサ間通信が発生しないので、通信のオーバーヘッドが大となることがない。

(7) 請求項7の発明は、LRU行列を用いて、作られた順が古く、かつ最近探索されていない葉から順番に削除対象として選べるようにしているので、デジタル探索

木で保持できる葉の数に制限がある場合にも、繰り返し探索される見出しの葉を削除しないで優先的に保持し続けて、削除と追加の処理が頻繁に発生することがないようにできる。

【図面の簡単な説明】

【図1】本発明の一実施形態のデジタル探索装置の構成図である。

【図2】請求項3に対応する第2の実施形態を説明するためのデジタル探索木の例を示す図である。

【図3】請求項4に対応する第3の実施形態を説明するためのデジタル探索木の例を示す図である。

【図4】請求項5に対応する第4の実施形態を説明するためのデジタル探索木の例を示す図である。

【図5】請求項6に対応する第5の実施形態を説明するためのデジタル探索木の例を示す図である。

【図6】請求項6に対応する第5の実施形態を説明するためのデジタル探索木の例を示す図である。

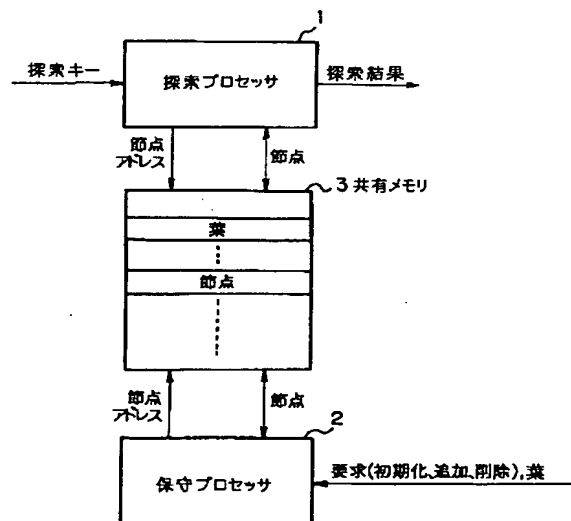
【図7】請求項7に対応する第6の実施形態を説明するためのデジタル探索木の例を示す図である。

【図8】従来技術で用いられているデジタル探索木の例を示す図である。

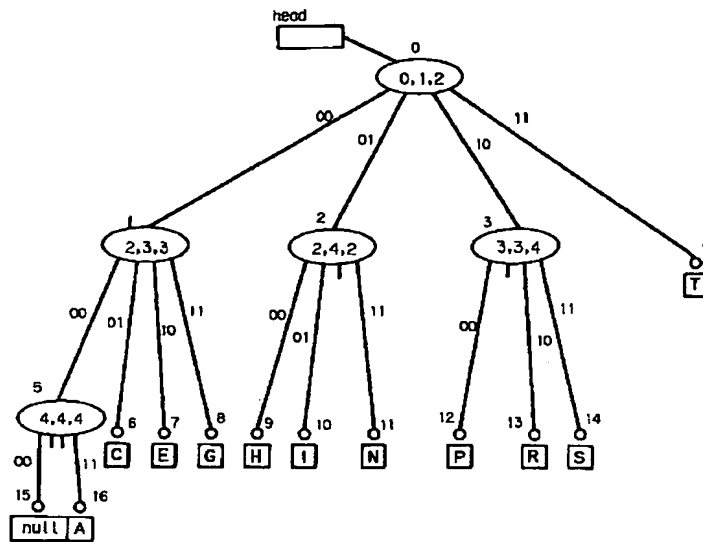
【符号の説明】

- 1 探索プロセッサ
- 2 保守プロセッサ
- 3 共有メモリ

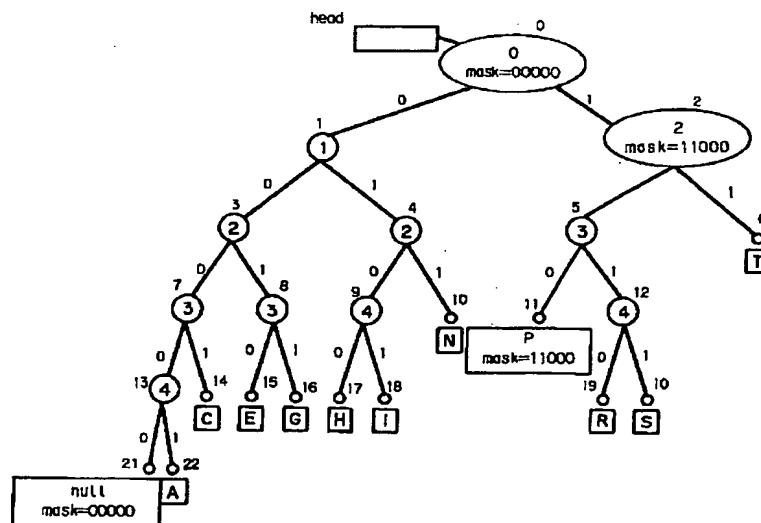
【図1】



【図2】

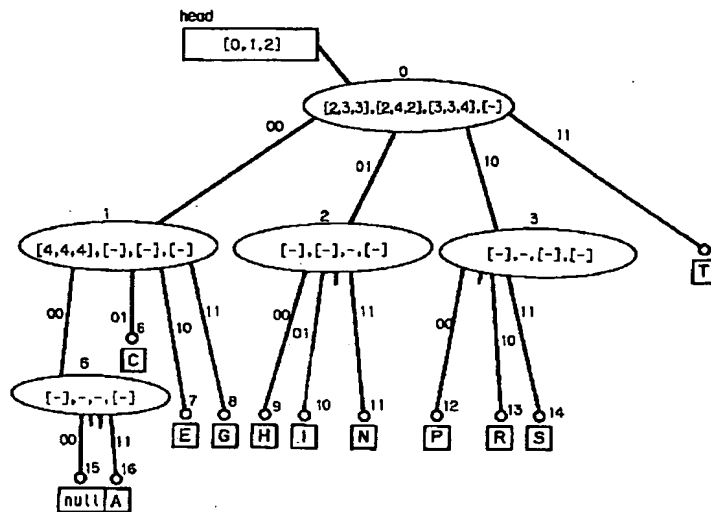


【図3】

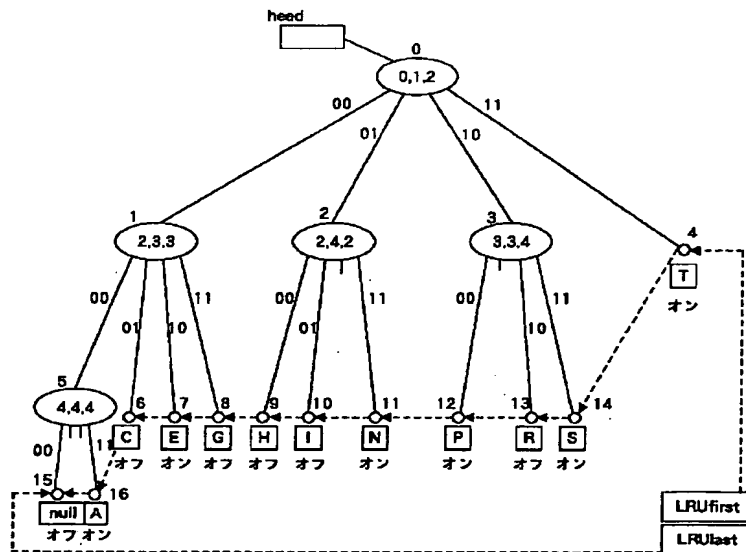


[illegible]

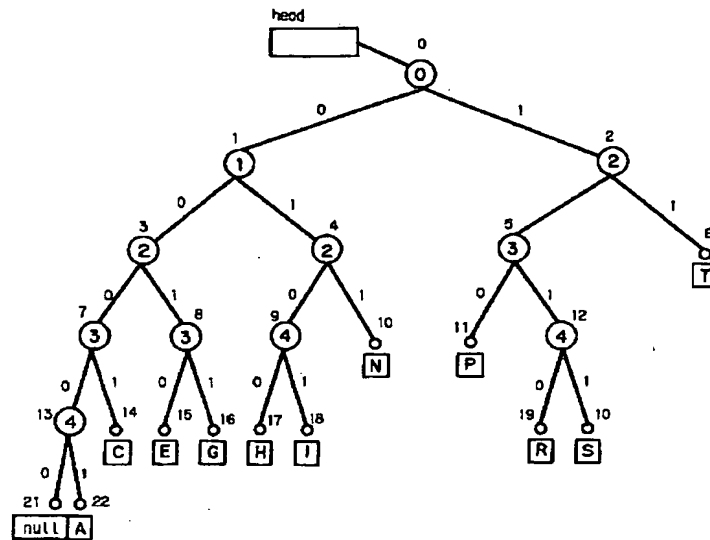
【図 6】



【図 7】



【図8】



フロントページの続き

(72)発明者 小倉 毅
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72)発明者 川野 哲生
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72)発明者 八木 哲
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内